A CORBA Gateway
Between the JTF-ATD and
the Distributed FactBase

by George W. Hartwig, Jr.
and Bruce McClure

ARL-TR-2024   August 1999

19991001 054

DTIC QUALITY INSPECTED 4

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

A CORBA Gateway Between the JTF-ATD and the Distributed FactBase

George W. Hartwig, Jr.
Information Science and Technology Directorate, ARL

Bruce McClure
Defence Science and Technology Organization
P.O. Box 1500
Salisbury, SA 5108
Australia

Abstract

Efforts to digitize the military have resulted in two worlds. The first is the high-echelon world of distributed processing, high-speed local area networks (LANs) and staff personnel. In this environment, distributed systems middleware, such as the Common Object Request Broker Architecture (CORBA), is playing an important role in command and control systems by providing a standardized backbone upon which to build and expand. In contrast, the low-echelon battlefield is characterized by low-bandwidth communications, little or no dedicated communications staff, and minimal computers. The software described in this report is a gateway to provide a degree of interoperability between the Distributed FactBase (DFB), a low-echelon system, and the Joint Task Force Advanced Technology Demonstrator (JTF-ATD) using the CORBA middleware standard. The gateway provides two CORBA interfaces to access the DFB. The first of these interfaces is a generic string-based interface, allowing access to any DFB commands and data. The second interface is specialized for receiving unit locations from the DFB whenever they are changed. The two CORBA interfaces to the DFB effectively build a CORBA wrapper for the DFB. In this report, we measure the throughput for the various interfaces and discuss the implications for the battlefield.

Table of Contents

INTENTIONALLY LEFT BLANK.

List of Figures

INTENTIONALLY LEFT BLANK.

1. Introduction

Efforts to digitize the military have resulted in two worlds. The first represents the high echelon world of distributed processing, high-speed local area networks (LANs) and staff personnel. In this environment, distributed systems middleware, such as the Common Object Request Broker Architecture (CORBA), is playing an important role in Command, Control, Communications and Intelligence (C3I) and related systems by providing a standardized backbone upon which to build and expand. On the other hand, the low-echelon battlefield is characterized by low-bandwidth communications, little or no dedicated communications staff and minimal computers. In this environment, voice traffic dominates, but more automated data communications are being introduced to gain new capabilities and support the increased tempo of the modern battlefield. This trend is often referred to as "digitizing the battlefield" and one of the major challenges of the digital battlefield is to seamlessly integrate these two disparate worlds. The software described in this report is an attempt to provide a gateway to move toward such an integration. By providing a degree of interoperability between the Distributed FactBase (DFB) and the Joint Task Force Advanced Technology Demonstrator (JTF-ATD) using the CORBA middleware standard [1], we hope to demonstrate some of the steps required in such an integration.

The fire advisor interface was orginally designed to connect a DFB to an expert system that ran on a Texas Instruments Explorer lisp machine. The fire advisor interface handled the details of communicating with the DFB and presented the expert system with a simple command and response string interface [2]. The gateway described in this report builds on this program by integrating the fire advisor interface with CORBA and providing two CORBA interfaces to the DFB. The first of these interfaces is a generic string-based interface, allowing access to DFB commands and data. The second interface is specialized for receiving unit locations from the DFB whenever they are changed. To demonstrate interoperability between the JTF mapserver and the DFB, we created a JTF specialist to display the information on the JTF browser. The two CORBA interfaces to the DFB effectively build a CORBA wrapper for the DFB. Thus, not only can the JTF-ATD use CORBA to communicate with the DFB but any CORBA-enabled program could be made to interoperate with the DFB through this gateway.

Why not run CORBA throughout the battlefield? The current generation of CORBA middleware has not been designed for low-bandwidth, error-prone communications channels. In fact, the most common protocol used for CORBA communications is the transmission control protocol (TCP), which is known to perform poorly in such an environment [3]. Further, the CORBA Internet Interoperability Protocol (IIOP) is a generic protocol that supports interoperability between different systems but adds significant overhead. In this report, we measure the throughput for the various interfaces and discuss the implications for the battlefield.

The following sections introduce the DFB, CORBA, and the JTF Map Server as background to understanding the rest of the report.

1

1.1   Overview of the DFB

The DFB is a prototype suite of software built by U.S. Army Research Laboratory (ARL) personnel to evaluate concepts for information distribution in tactical digital networks [4]. The software at each battlefield node* is called a DFB and is composed of a security control module (SCM), a factbase (FB) and various interface modules. The SCM is where incoming fact base commands are examined and scheduled for processing that may include storage or forwarding to other nodes. This software also provides an "active trigger" capability in response to requests from application programs co-located† with the DFB. To implement this scheme while allowing maximum flexibility, a generic set of software modules was written, and these are configured at run time by processing a CAPability (CAP) profile file. Included in the CAP file is information describing the communications environment in which this node would be operating and a set of rules that would control information forwarding to other nodes. Figure 1 shows the overall DFB architecture. The rule concept allows a single program to be designed so that when combined with the appropriate CAP file, it meets the operational communication requirements of any combat unit in the brigade and below tactical scenario. The prototype software described in this report is written in the C programming language and currently runs under the SOLARIS 2.6 operating system on a variety of SUN computer systems.
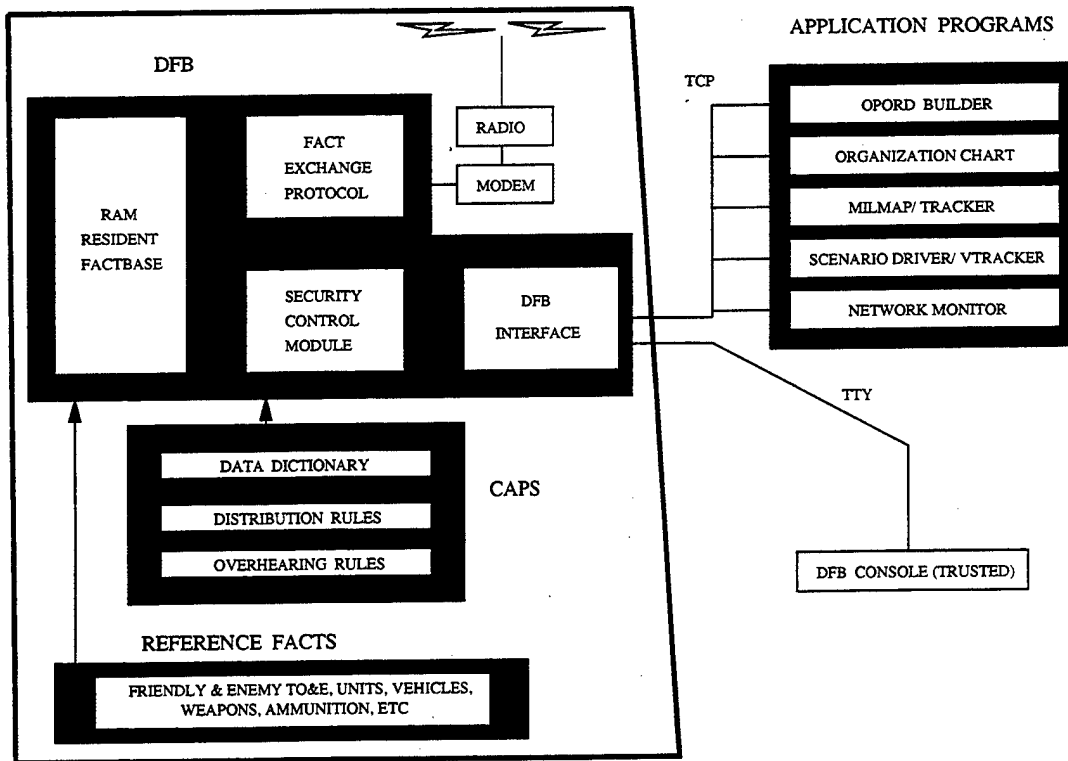
Figure 1. DFB Software Architecture.

---

*For our purposes here, a node is any unit or vehicle with a computer and a radio.

†Co-located means on the same computer or connected with a high-speed, reliable protocol, such as TCP.

1.2 Overview of CORBA

The CORBA [1] can be characterized as object-oriented middleware. This middleware is a layer of software and services that support interactions between software objects on hetrogeneous platforms. The CORBA specification was defined by the Object Management Group (OMG), an industry consortium. An important part of the CORBA middleware is the Object Request Broker (ORB). Figure 2 depicts the relationships between the ORB and client and server objects.
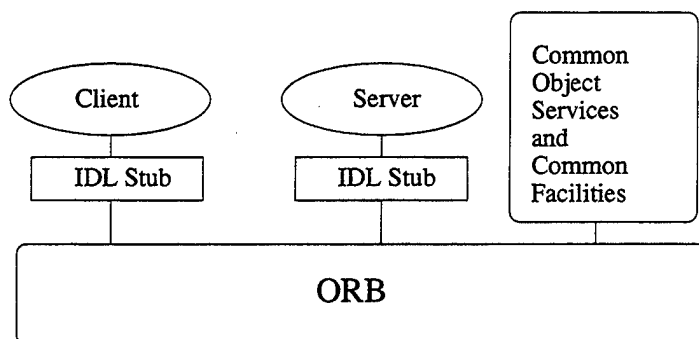
Figure 2. CORBA Architecture.

A key feature of CORBA is the Interface Definition Language (IDL). CORBA IDL is a descriptive language based on C++ syntax. An IDL specification defines a contract between object services (servers) and client objects. The compiled IDL is stored in the CORBA Interface Repository within the ORB and used to ensure type safety between objects at runtime. Since requests to a CORBA service can be created dynamically, using the Dynamic Invocation Interface (DII), it is possible for a client object to make use of a service that did not exist when the client was compiled–and this with type safety! This ability to flexibly combine heterogeneous objects into new applications is what makes CORBA such a useful technology.

The ability of CORBA to cope with heterogeneity comes in several ways. First, IDL can be mapped to different languages, meaning that clients and servers can be written in different programming languages and no extra effort is required to make them interoperate. Second, CORBA's Remote Procedure Call mechanism takes care of differences in computing platforms (such as byte orderings). Finally, CORBA can be run over a variety of communications protocols.

According to a recent MITRE report [5], CORBA is the middleware of choice for 1998/1999.

1.3 Overview of the Joint Task Force Advanced Technology Demonstrator

The JTF-ATD is an ongoing Defense Advanced Research Projects Agency (DARPA) project aimed at demonstrating cutting edge technology for use in CORBA-based C3I sys-

3

tems [6]. The JTF-ATD has a number of servers integrated with displays using CORBA. The service of interest for the gateway is the Mapserver. The Mapserver is an open system for displaying map-related information. Developers are able to extend the Mapserver by adding new layers to the map. New layers can be added by implementing the JTF-ATD "specialist interface."
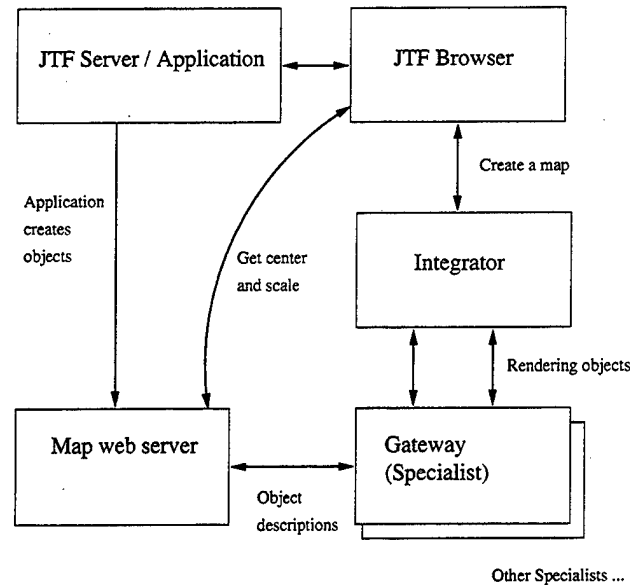

Figure 3. JTF Mapserver Architecture.

Figure 3 depicts a simplified process model of the Mapserver. For the gateway described in this report the important interactions are between the specialist and the integrator. These interactions occur when a map is being created or resized, and the specialist layer is to be displayed. Note that there are multiple specialists-the JTF software comes with several complete specialist layers (e.g., roads). To demonstrate the capabilities of the gateway, we implemented a new but simple specialist that displays unit information received from the DFB via the gateway.

2. Architecture of the Gateway

The purpose of the gateway is to provide access to the DFB from CORBA and ultimately to allow the JTF-ATD to interact with the DFB. The mechanism used by the gateway to interact with the DFB is the package protocol (PKG) interface. Figure 4 depicts the architecture of the gateway.

The architecture follows the style of a classic three-tier model: application server on computer A, application logic or middle tier on computer B, and application client on computer C. Figure 4 depicts these programs running on three separate computers, however, this need not be the case. In addition, existence of a CORBA ORB on computers B and C is not shown in the figure.
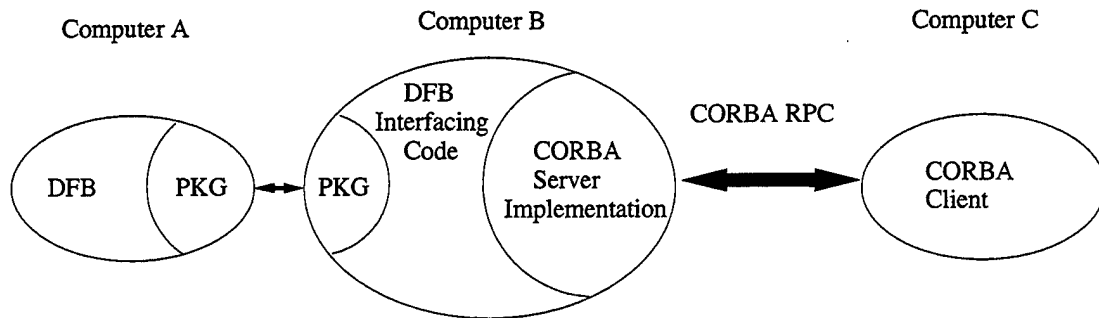
4

Figure 4. Gateway Architecture.

On the DFB side of the gateway, there are four main operations: open a connection to the DFB, close the connection to the DFB, pass a command to the DFB, and request outstanding triggers. DFB commands are embedded in strings and sent to the DFB, which returns a string as a result. Refer to Hartwig [7] for the detailed syntax and semantics of DFB commands. The functionality of the request trigger operation is more complex. A calling program is able to install a trigger in the DFB, which will fire upon a specified condition such as an update to a fact in the DFB. Triggers can be set up in the DFB using a DFB command such as the following:

trigger "grid_trigger" grid (1);

This trigger is designed to fire whenever any fact of type "grid" is updated. * Within the DFB, the firing of a trigger is handled immediately. All notifications are sent after the fact is updated. Those applications that have requested the notifications will receive unexpected data or asynchronous communications after the fact is updated. The gateway, however, can either store the results of a trigger firing until the CORBA client requests trigger information (generic string interface) or immediately send the data to the client (JTF unit interface). The generic interface allows the CORBA client to use normal blocking remote procedure call (RPC) semantics (client calls server, waits for a response and then continues). Thus, when a client requests trigger information, it may receive zero, one or many sets of results embedded in a single string.

On the CORBA side of the gateway, the CORBA server supports four operations which mimic those on the DFB side.

---

*The "1" is an expression value. An expression that evaluates to a nonzero value is said to have returned a "true" value, so by using a constant "1" here, we are asking for notification whenever a grid fact is altered or created.

5

3. The CORBA Interface

3.1 Generic String Interface

The generic string interface offered by the gateway can best be understood by looking at the IDL:

```
interface jfbg {
        // IDL operations
        //
        // Return codes are defined in jfbgh.h
        // Open a connection to DFB for this client and return code
        short open();
        // We are done, so close the connection to the DFB and
        // cleanup any memory.
        short close();
        short request_triggers(out string triggers);
        short dfb_command(in string cmd, out string response);
};
```

These four operations return a status code to indicate success or otherwise. In addition, these operations may throw a CORBA exception in the event of a system error (e.g., communications error). The open command should be called prior to any other commands–this sets up communications with the DFB. Most of the work is done by issuing a dfb_command call. The dfb_command passes a string to the DFB and returns a string result. This is a very flexible interface that supports all kinds of DFB operations. The request_triggers operation is used to gather any results from triggers that have fired since the last request. The results are returned in a single string. Finally, the close operation tells the gateway to close its connection to the DFB.

3.2 JTF Unit Interface

This interface is designed specifically to support a JTF-ATD Mapserver application. There are five relevant operations–open, close, signOn, signOff, and send_units that are defined in an extension of the previous interface and introduction of a new interface:

```
interface JFBGUnit {
        // Send a solitary unit back to client
        oneway void send_units( in long unit_id,
                        in float lat,
                        in float lon);
};

interface jfbg {
```

```
        // Open a connection to DFB for this client and return code
        short open();
        short close();
        short request_triggers(out string triggers);
        short dfb_command(in string cmd, out string response);
        // signal that a client is ready for callbacks
        void signOn(in JFBGUnit CallObj);
        // signal that a client does not want callbacks anymore
        void signOff(in JFBGUnit CallObj);
};
```

The gateway initiates a send_units RPC to any registered clients whenever a trigger is fired. Open and close affect the connection of the gateway to the DFB in the same manner as the generic interface. SignOn and signOff are used by the client to indicate that it is ready to receive callbacks from the server. The send_units RPC is a CORBA callback, which contains the identifier (ID), latitude and longitude of a given unit. The client program must be able to deal with reports of units with the same unit ID (e.g., position updates). Our JTF specialist handles this is by overwriting the previous positions.

3.2.1   Coordinate Conversions

When interfacing systems together, not only must the physical and data link protocols be considered but any data mapping required must be performed. Our scenario is no exception. The unit location data input to the DFB comes from ModSAF (Modular Semi-Automated Forces). ModSAF produces Universal Transverse Mercator (UTM) coordinates. The JTF Mapserver software we are using requires latitudes and longitudes. To perform this translation, a modification of the U.S. Geological Survey (USGS) Program J380 was used. This program was originally written in FORTRAN and was intended to be a stand-alone executable in a batch or background environment. This adaptation is written in C and is intended to be linked with other routines, which will call it for one point conversion at a time. The original FORTRAN version is dated 14 August 1984.

The software used here is a "C" translation of the original FORTRAN and consists of the following files.

ll_conv - A user interface that accepts latitude and longitude data in a DDDMMSSH format and converts it into a zone number and a six-digit easting and a seven-digit northing.

utm_conv - A user interface that prompts for and accepts UTM coordinates consisting of a zone, easting, and northing. It then returns a latitude and longitude in the format DDD:MM:SS.SSSS (degrees:minutes:seconds).

utmc - This routine is called by both of the above programs to perform the actual conversions. The Clark 1866 spheroid is used by default, although others are available for use. Southern-hemisphere latitudes and eastern-hemisphere longitudes are considered negative, and must be input as such. *Note that this is nonstandard for longitude.*

ll_conv_sub - This file provides a programming interface to the utmc function. LAT-LONG_TO_UTM takes a latitude and longitude in degrees and converts it to a UTM zone, easting, and northing. UTM_TO_LATLONG takes a UTM zone, easting, and northing and converts it to a latitude and longitude in degrees.

Brief descriptions of various coordinate systems including UTM may be found at Peter Dana's web page at the University of Texas at Austin [8].

4. Performance Analysis

In the introduction, we discussed the importance of bandwidth usage on the battlefield. Following development of the gateway, the throughput was measured for all the TCP connections related to the gateway. A DFB to DFB link was included so that the effects of low-bandwidth techniques could be evaluated.

4.1 Configuration

Figure 5 shows the experimental configuration we would have used if time and money were not obstacles. Here we would have four vehicles, each equipped with a Global Positioning System (GPS) receiver, a computer and a single channel ground/air radio system (SINCGARS). They would move around a test area on preplanned routes and using principles of model-based information distribution (see the Appendix), each vehicle would send updates as required to a central location represented in the figure as the Gateway Computer. These updates would be used by the route modelers to modify the models they are using to predict locations. Finally, predicted locations would be sent via the DFB to the gateway program and entered into the CORBA and JTF world. Since time and cost were considerations, Figures 6 and 7 show the two-part setup that was actually used to take measurements.

In Figure 6 the four boxes on the left side in the oval labeled "Source Computer" represent programs that play back time-tagged route data to the DFB. These replace the vehicles in Figure 5. The ModSAF program* generated this route data. These programs could be replaced with a direct connection to the ModSAF program. This would allow the direct interface of ModSAF to JTF-ATD software via the DFB. Work in support of such an effort has already been completed by Howell Caton of ARL and is described in Caton [9]. The boxes named "model player n" represent models of the remote unit routes. They predict the routes using the planned routes and use the actual position updates to update the planned

---

*"ModSAF (Modular Semi-Automated Forces) is a set of software modules and applications used to construct Advanced Distributed Simulation (ADS) and Computer-Generated Forces (CGF) applications. ModSAF modules and applications let a single operator create and control large numbers of entities that are used for realistic training, test, and evaluation on the virtual battlefield. ModSAF contains entities that are sufficiently realistic resulting in the user not being aware that the displayed vehicles are being maneuvered by computers, rather than human crews. These entities, which include ground and air vehicles, dismounted infantry (DI), missiles, and dynamic structures, can interact with each other and with manned individual entity simulators to support training, combat development experiments, and tests of evaluation studies" [10].
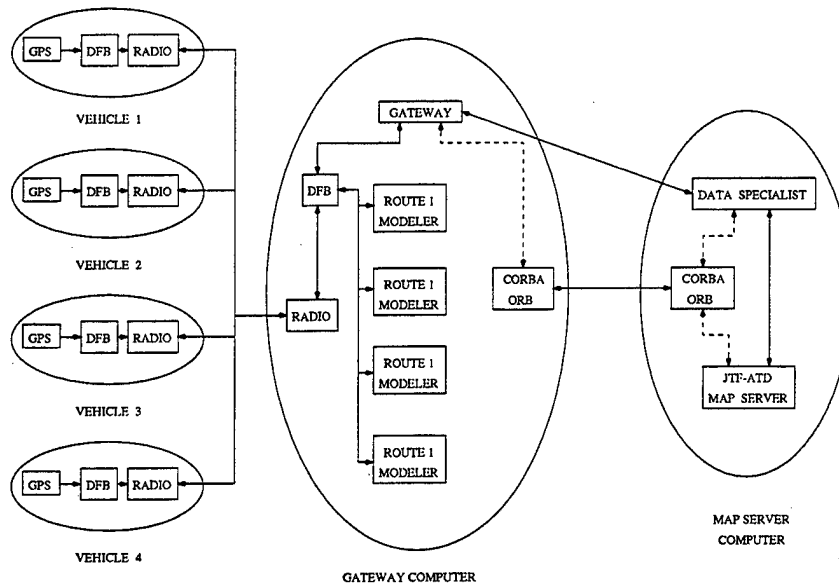
8

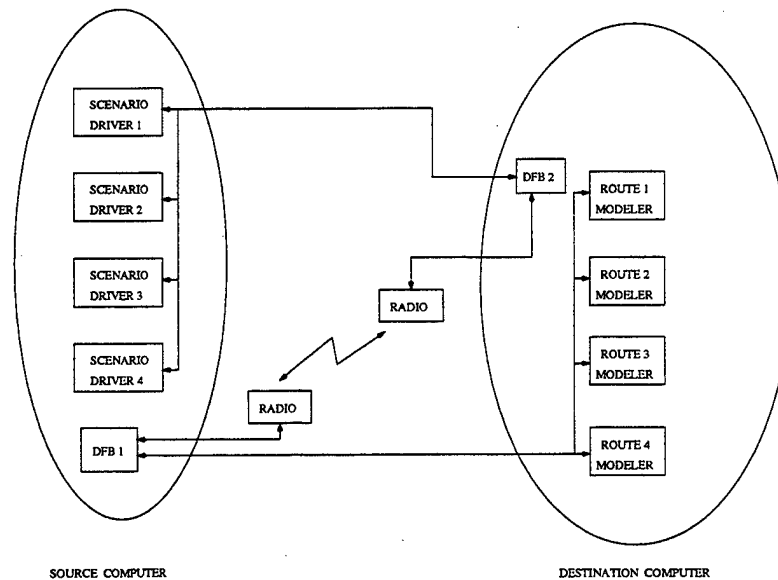**Figure 5. Desired Experimental Configuration.**

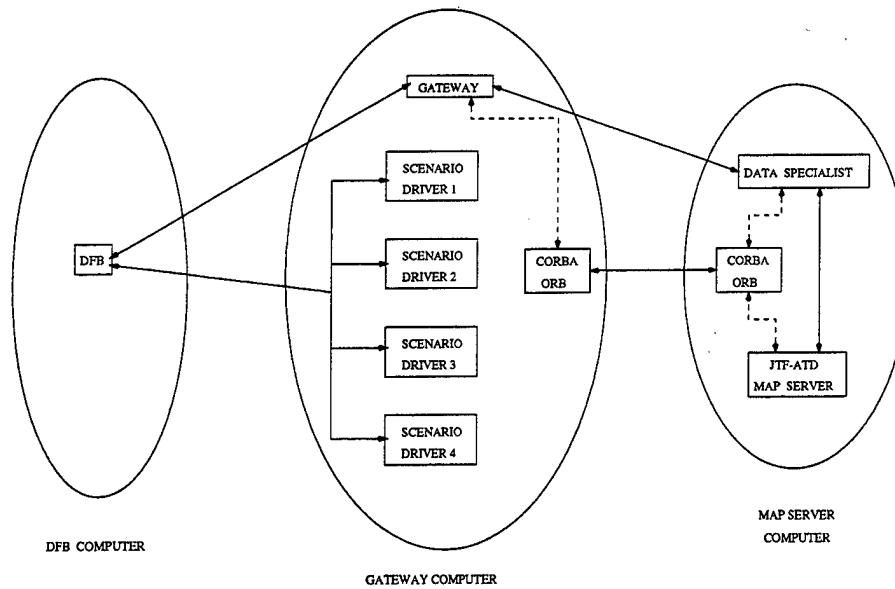**Figure 6. Experimental Setup (Part 1).**

Figure 7. Experimental Setup (Part 2).

routes. The DFB boxes represent programs that were described earlier in this report.

This DFB-to-DFB link is included to demonstrate the advantages of Model-based communications. For more details see the Appendix and the references at the end of this paragraph. Studies show that this technique is very effective in reducing bandwidth while continuing to convey a satisfactory level of information [11,12].

The somewhat circular arrangement of application programs and DFBs was the result of using the tcpdump program to collect communications information. Tcpdump cannot capture information that stays on a single machine; therefore, all connections were from one computer to another. The tcpdump package [13] was used to capture all TCP traffic between the three nodes in Figures 6 and 7. The order of program startup for the measurements was DFB, playback, gateway, and JTF. The tcpdump program was started after the playback programs to enable a meaningful comparison. In fact, all data prior to the start of the JTF specialist were removed from the file. The tcpdump program was stopped while all the programs were running. After the data were recorded, data from unrelated connections were filtered out. Then a special-purpose C program was used to total the packets and bytes for each connection.

4.2   Performance Measurement Results

The results of the measurements are summarized in Figures 8, 9, and 10. At first glance, these results seem suprising: Why is CORBA using so little bandwidth?

In comparing the throughput from each set of connections, consider the following points:

- Data in the CORBA connections go one way, because the JTF unit interface uses the CORBA one-way RPC feature.
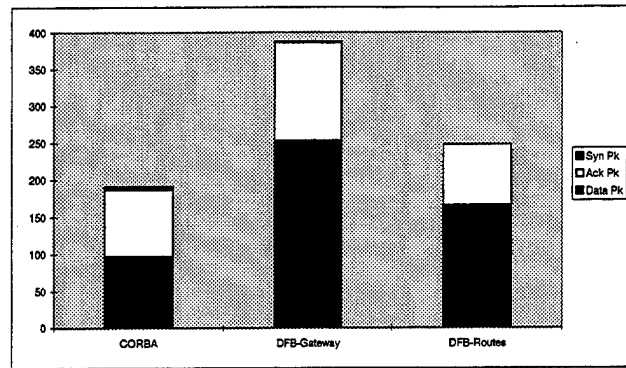
10

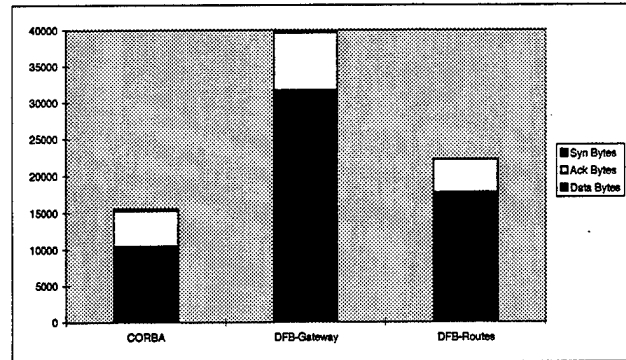Figure 8. Measured TCP Traffic by Packets.

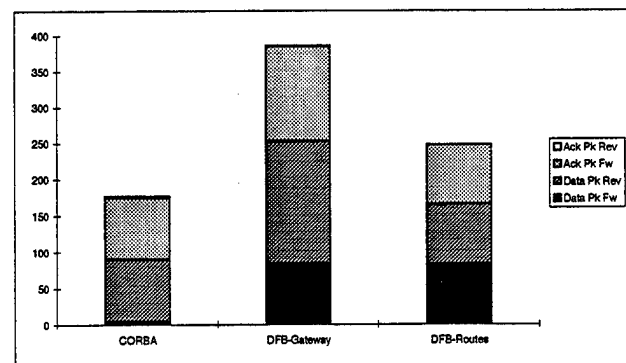Figure 9. Measured TCP Traffic by Bytes.

Figure 10. Measured TCP Traffic by Packets and Direction.

11

- The DFB trigger mechanism requires the application to get the data associated with a trigger in a separate request.

- The measurements were recorded in a high-bandwidth, low-error rate network so that TCP performance is good.

- The model-based communications that the DFB is capable of is not being used here because the gateway traffic would not be expected to run over low-bandwidth links.

Notice that the traffic across the radio link is a fraction of all other traffic. Observation of this measurement shows just how much bandwidth can be saved using a model-based approach.

When one designs distributed applications for use on wire-based LANs, the bandwidth available between components is almost never a consideration except in the most extreme cases, i.e., video teleconferencing. When the environment includes wireless links, priorities must be realigned. These wireless links have very different characteristics than the hardwired, fixed-host, land-based networks. These fibre and wired links have Bit Error Rates (BER) in the range 10^{-10} and 10^{-5}, which are significantly different from 10^{-3} or worse, typical of radio links. Careful use of radio links in a tactical enviroment is very important – more so because of competition for the same frequencies inherent in the low-level communication protocols. It is with this in mind that the results depicted in Figures 11 and 12 are significant.
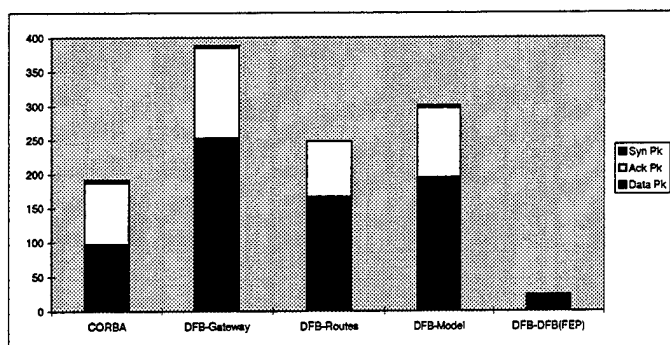

Figure 11. Measured TCP Traffic by Packets.

5.   Conclusions

One of the major challenges of the digital battlefield is to seamlessly integrate the high-echelon, high-bandwidth with the low-echelon, low-bandwidth world to meet mission objectives. We have described a gateway that provides two CORBA interfaces to access the DFB. The first of these interfaces is a generic string-based interface, allowing access to any DFB
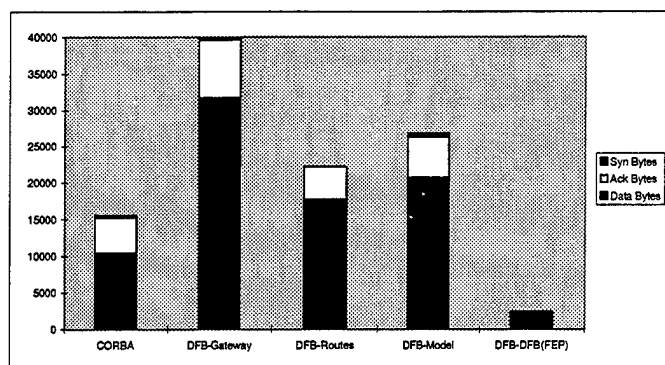
Figure 12. Measured TCP Traffic by Bytes.

commands and data. The second interface is specialized for receiving unit locations from the DFB whenever they are changed. The two CORBA interfaces to the DFB effectively build a CORBA wrapper for the DFB. Thus, not only can the JTF-ATD use CORBA to communicate with the DFB, but any CORBA-enabled program could be made to interoperate with the DFB through this gateway.

Measurements of the number and size of packets flowing between the various components (e.g., route generators, DFBs, trajectory models, the JTF Mapserver and the CORBA gateway) demonstrate that model-based information distribution can significantly reduce bandwidth requirements.

In summary then, we have demonstrated, on a small scale, technology that can assist with the problems of interoperability and connectivity on the battlefield and with higher command.

INTENTIONALLY LEFT BLANK.

6. References

1. Common Object Request Broker Architecture and Specification, Revision 2.0, OMG, July 1995.

2. Kaste, Richard C. "Fireadvisor User's Manual," Technical Report TR-3219, U.S. Army Ballistic Research Laboratory, APG, MD, March 1991.

3. Lui, Y.J. "The Performance of TCP in a Tactical Environment," IEEE MILCOM 97, 1997.

4. Chamberlain, Samuel C., "The Information Distribution System: IDS - An Overview," Technical Report TR-3114, U.S. Army Ballistic Research Laboratory, APG, MD, July 1990.

5. MITRE. "Recommendations for Using DCE, DCOM and CORBA Middleware," 13 April 1998.

6. Mckenna, Christine, and Rick Hayes-Roth. "Introduction to the JTF Reference Architecture," Tecknowledge, October 1996.

7. Hartwig, George W. "The Information Distribution System: the Factbase," Technical Report TR-3247, U.S. Army Ballistic Research Laboratory, APG, MD, July 1991.

8. Dana, Peter H. World Wide Web Page:
   http://www.utexas.edu/depts/grg/gcraft/notes/coordsys/coordsys.html
   Department of Geography, University of Texas at Austin, revised 18 Aug 1998.

9. Caton, Howell. "A Demonstration of a Computationally Intensive Situation Awareness Methodology," Technical Report, U.S. ARL, APG, MD, to be published.

10. World Wide Web Page:
    http://www.stricom.army.mil/STRICOM/E-DIR/ES/MODSAF/Project
    Office: Engineering Directorate, Synthetic Environment and Technology Management Division (E-DIR) .

11. Smith, Kenneth G., and Eric G. Heilman, "Information Distribution Technology Applied to Army Aviation and Air Defense Counter-Air Operations," BRL-TR-3397, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, September 1992.

12. Hartwig, George W. Jr., Frederick S. Brundick, and CPT(P) Scott D. Kothenbeutel. "Tactically Significant Route Planning," ARL-TR-1139, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, September 1992.

13. Lawrence Berkeley National Laboratory, Network Research Group, tcpdump@ee.lbl.gov, ftp://ftp.ee.lbl.gov/tcpdump.tar.Z, 27 January 1998.

INTENTIONALLY LEFT BLANK.

Appendix:

Model-Based Situational Awareness

INTENTIONALLY LEFT BLANK.

Model-based communications is a paradigm for the distribution of digital information over low-bandwidth channels, particularly combat net radios. With this technique, each node maintains a more or less sophisticated model of the battlefield events depending on available computer power, need, etc. When applied to the problem of situational awareness, each node has a copy of everybody's planned route. On the sending node the planned route is compared to the actual route being traversed and if the difference exceeds a predetermined threshold, an update is sent to the receiving node. The receiver uses the planned route to predict locations. When an update is received, it is added to the planned route and then the model continues to predict the sending node's location using this updated information.

Figure A-1 shows planned routes for the four vehicles used in this demonstration. These routes each contain seven points each with time tag. The model then consists of these points, or more accurately, the six line segments they define, a search engine to find the appropriate segment to use and an interpolation function. When furnished a time, the planned route is searched to determine the appropriate segment. Then the time and the segment are passed to the interpolation routine so that the location corresponding to the time may be determined.

Figure A-2 shows the routes actually traversed as represented by the two second updates generated by the ModSAF simulation and fed to the demo by the scenario drivers. These routes have 322 points for vehicle 11, 337 points for vehicle 12, 331 points for vehicle 13, and 331 points for vehicle 14. The final figure, Figure A-3, shows the planned routes with the updates included. These routes include 9 points for vehicle 11, 10 points for vehicle 12, 11 points for vehicle 13, and 10 points for vehicle 14.
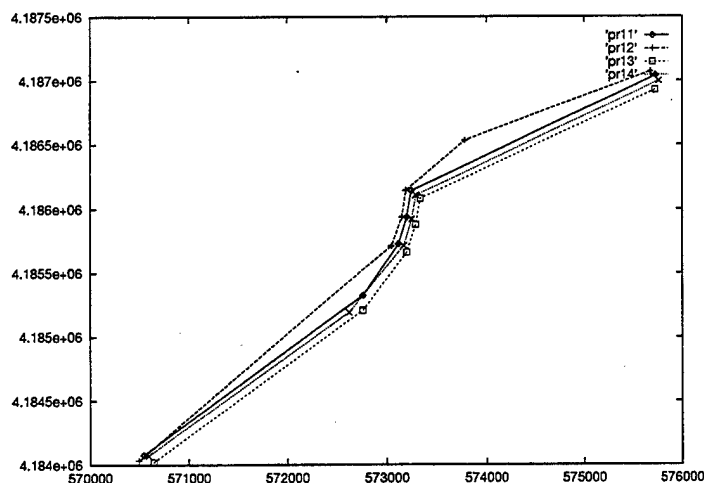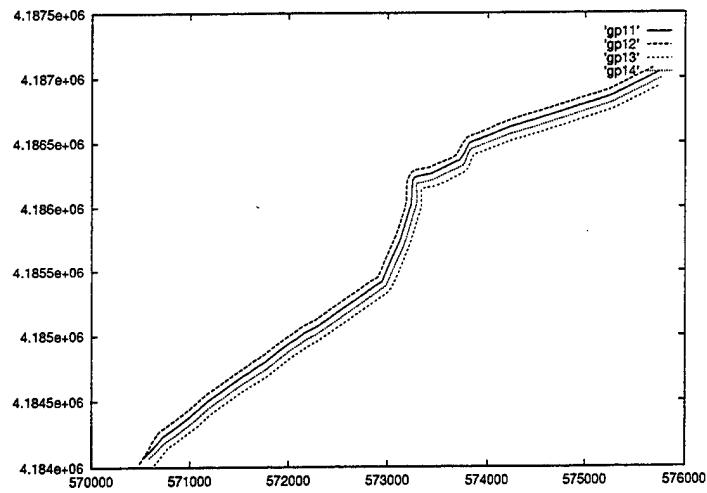

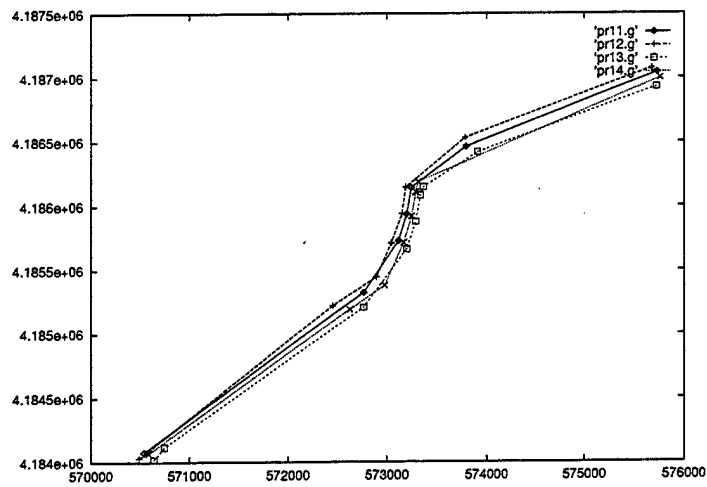**Figure A-1. Planned Route.**

Figure A-2. Actual Route.

Figure A-3. Altered Route.

20

Glossary

ARL	U.S. Army Research Laboratory
ADS	Advanced Distributed Simulation
ATD	Advanced Technology Demonstration
BER	Bit Error Rate
BRL	U.S. Army Ballistic Research Laboratory
CAP	CAPability file
C3I	Command, Control, Communications, and Intelligence
CGF	Computer-Generated Forces
CORBA	Common Object Request Broker Architecture
DARPA	Defense Advanced Research Projects Agency
DFB	Distributed Fact Base
DI	Dismounted Infantry
DII	Dynamic Invocation Interface
FB	Fact Base
GPS	Global Positioning System
ID	Identifier
IDL	Interface Definition Language
IDS	Information Distribution System
IIOP	Internet Inter-ORB Protocol
JTF	Joint Task Force
JTF-ATD	Joint Task Force Advanced Technology Demonstrator
LAN	Local Area Network
ModSAF	Modular Semi-Automated Forces
ORB	Object Request Broker
OMG	Object Management Group
PKG	Package Protocol
RPC	Remote Procedure Call
SCM	Security Control Module
SINCGARS	Single Channel Ground/Air Radio System
STRICOM	Simulation, Training, and Instrumentation Command
TCP	Transmission Control Protocol
TR	Technical Report
USGS	U.S. Geological Survey
UTM	Universal Transverse Mercator

INTENTIONALLY LEFT BLANK.

NO. OF COPIES	ORGANIZATION	NO. OF COPIES	ORGANIZATION
2	DEFENSE TECHNICAL INFORMATION CENTER DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218	1	DIRECTOR US ARMY RESEARCH LAB AMSRL DD J J ROCCHIO 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	HQDA DAMO FDQ D SCHMIDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460	1	DIRECTOR US ARMY RESEARCH LAB AMSRL CS AS (RECORDS MGMT) 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	OSD OUSD(A&T)/ODDDR&E(R) R J TREW THE PENTAGON WASHINGTON DC 20301-7100	3	DIRECTOR US ARMY RESEARCH LAB AMSRL CI LL 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DPTY CG FOR RDE HQ US ARMY MATERIEL CMD AMCRD MG CALDWELL 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001		ABERDEEN PROVING GROUND
1	INST FOR ADVNCD TCHNLGY THE UNIV OF TEXAS AT AUSTIN PO BOX 202797 AUSTIN TX 78720-2797	4	DIR USARL AMSRL CI LP (305)
1	DARPA B KASPAR 3701 N FAIRFAX DR ARLINGTON VA 22203-1714		
1	NAVAL SURFACE WARFARE CTR CODE B07 J PENNELLA 17320 DAHLGREN RD BLDG 1470 RM 1101 DAHLGREN VA 22448-5100		
1	US MILITARY ACADEMY MATH SCI CTR OF EXCELLENCE DEPT OF MATHEMATICAL SCI MAJ M D PHILLIPS THAYER HALL WEST POINT NY 10996-1786		

NO. OF
COPIES ORGANIZATION

ABERDEEN PROVING GROUND

1   COMMANDER
        US ARMY ERDEC
        SCBRD TD J VERVIER
        APG MD 21010-5423

19   DIR USARL
        AMSRL IS T
          S CHAMBERLAIN
        AMSRL IS TP
          B COOPER
          C RETTER
          F BRUNDICK
          L MARVEL
        AMSRL IS CI
          B BROOME
          H CATON
          G HARTWIG (10 CPS)
          M LOPEZ
          A BRODEEN

NO. OF
COPIES ORGANIZATION

1   DSTO CE RESEARCH CENTRE
     FERNHILL PARK
     DEPARTMENT OF DEFENCE
     CANBERRA ACT 2600
     DR IAN CHESSEL DESRL
     MR NEIL BRYANS CITD
     DR WARREN HARCH RLC3
     MR JOHN MANSFIELD HDSG
     DR ANTHONY BEDROD CCD

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1999	Final, May 98 - Oct 98

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
A CORBA Gateway Between the JTF-ATD and the Distributed FactBase	
6. AUTHOR(S)	622783.Y10
George W. Hartwig, Jr. and Bruce McClure*	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Research Laboratory ATTN: AMSRL-IS-TP Aberdeen Proving Ground, MD 21005-5067	ARL-TR-2024

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)	10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

*Defence Science and Technology Organization, P.O. Box 1500, Salisbury, SA 5108, Australia

12a. DISTRIBUTION/AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited.	

13. ABSTRACT *(Maximum 200 words)*

Efforts to digitize the military have resulted in two worlds. The first is the high-echelon world of distributed processing, high-speed local area networks (LANs) and staff personnel. In this environment, distributed systems middleware, such as the Common Object Request Broker Architecture (CORBA), is playing an important role in command and control systems by providing a standardized backbone upon which to build and expand. In contrast, the low-echelon battlefield is characterized by low-bandwidth communications, little or no dedicated communications staff, and minimal computers. The software described in this report is a gateway to provide a degree of interoperability between the Distributed FactBase (DFB), a low-echelon system, and the Joint Task Force Advanced Technology Demonstrator (JTF-ATD) using the CORBA middleware standard. The gateway provides two CORBA interfaces to access the DFB. The first of these interfaces is a generic string-based interface, allowing access to any DFB commands and data. The second interface is specialized for receiving unit locations from the DFB whenever they are changed. The two CORBA interfaces to the DFB effectively build a CORBA wrapper for the DFB. In this report, we measure the throughput for the various interfaces and discuss the implications for the battlefield.

14. SUBJECT TERMS		15. NUMBER OF PAGES
distributed computing, CORBA, low bandwidth		27
		16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author____ARL-TR-2024 (Hartwig)_____ Date of Report _August 1999_____

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

_____

_____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

_____

_____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

_____

_____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

_____

_____

_____

CURRENT
ADDRESS

_____

Organization

_____

Name   E-mail Name

_____

Street or P.O. Box No.

_____

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD
ADDRESS

_____

Organization

_____

Name

_____

Street or P.O. Box No.

_____

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)